2013

Towards Automated Analysis of Business Processes for Financial Audits

Michael Werner
University of Hamburg, Germany, michael.werner@wiso.uni-hamburg.de

Nick Gehrke
NORDAKADEMIE, Elmshorn, Germany, nick.gehrke@nordakademie.de

Markus Nüttgens
University of Hamburg, Germany, markus.nuettgens@wiso.uni-hamburg.de

Towards Automated Analysis of Business Processes for Financial Audits

Michael Werner[1], Nick Gehrke[2], and Markus Nüttgens[1]

[1] University of Hamburg, Germany
{michael.werner, markus.nuettgens}@wiso.uni-hamburg.de
[2] NORDAKADEMIE, Elmshorn, Germany
nick.gehrke@nordakademie.de

Abstract. Financial audits play a significant role in the economy by safeguarding the correctness of published financial information. Public auditors face the challenge to audit financial statements that are created by increasingly integrated and complex information systems. This paper addresses a specific problem in the auditing process. A major challenge in this process is the analysis and audit of business processes that produce financial entries. We illustrate results from applying business process mining techniques to extensive test and real life data and discuss gained insights from the application for the development of automated business process analysis methods in the context of financial audits.

Keywords: Process Mining, Financial Audits, Business Process Analysis

1 Introduction

Financial audits play a significant role in modern economies. Companies publish financial statements in order to inform relevant stakeholders. For preventing misinformation of the addressees financial statements are subject to audits that are mandated by law and specified in regulatory requirements. The audits are carried out by public auditors who follow specific audit approaches for planning and executing their audits. Audit standards require that auditors consider and test relevant business processes during the audit [1]. The requirement derives from the assumption that well controlled transaction processing will lead to valid entries on the balance sheet and profit and loss statements. When business transactions are carried out in a correct and controlled manner they will most likely lead to complete and accurate journal entries.

With increasing integration of the execution of business processes in information systems and the accompanied progress in automation of transaction processing it becomes more and more challenging to audit these processes. Contemporary audit approaches take into account the relevance of business processes, supporting information systems and internal control frameworks, but they basically rely on manual audit procedures to analyze and test them. The manual procedures primarily include interviews for obtaining information and manual test activities for evaluating relevant controls. With increasing integration of information systems for supporting and auto-

11th International Conference on Wirtschaftsinformatik,
27th February – 01st March 2013, Leipzig, Germany

mating transaction processing audit activities like interviews and manual audit activities become inefficient or even ineffective due to the increasing complexity and the mere volume of processed transactions [2].

An alternative would be the application of automated analysis and audit procedures as a business intelligence tool that supports the auditor in the auditing process. [3] conceptually illustrate how process mining methods can be combined with automated application control testing methods for designing automated audit methods. A requisite for such a development are methods that allow an automated analysis of business processes. The analysis results can then be used for automated testing purposes.

When information systems are used to support or automate the transaction processing they also provide information that can be used for an automated analysis. By using process mining techniques [4] and specific mining algorithms for financially relevant business processes [5] executed process instances can be mined, reconstructed and analyzed.

In this paper we focus on the aspect of automated analysis. We apply an existing mining algorithm for financially relevant business processes to test data and real life data. The aim of this research is to evaluate which insights can be derived by analyzing the application of the implemented algorithm. We statistically analyze the mined business processes instances that are reconstructed from the available data to identify which further research and improvement is needed on the path towards automated analysis methods.

We start with an illustration of related work in section two, followed by a brief description of the applied research methodology in section three. The mined process instances are represented as Petri nets. The used representation, the chosen mining method and the experimental setup are explained in section four. Section five provides the results from analyzing the process instances that were mined from the test and real life data. A discussion of the gained results and an illustration of identified limitations followed by a brief summary and conclusion close the paper.

2 Related Work

Of particular interest for the research laid out in this paper are publications from the field of process mining. Research on process mining started in the late 1990s by [6] and has gained extensive attention in the last decades. Significant research work has been published by van der Aalst et al. leading to a comprehensive basic publication on process mining that covers major aspects of the research domain [4].

From a financial accounting and auditing perspective requirements are outlined in relevant audit standards. The major international standard setting body is the International Auditing and Assurance Standards Board (IAASB) which publishes the International Standards on Auditing (ISA). The ISA 315 (Revised) “Identifying and Assessing the Risks of Material Misstatement through Understanding the Entity and Its Environment“ outlines the requirement to consider business processes and related internal controls in order to assess the risk for material misstatement (ISA 315.18) [1].

The role of information systems for accounting is well researched but few authors address the role of information systems in the context of auditing. [7] describes techniques to audit enterprise resource planning (ERP) systems, but the exploitation of information that is available in information systems for the purpose of automated analyses is a relatively novel field of research as illustrated by [8].

Specific research on process mining for auditing purposes has gained increased attention over the last two to three years. [9] offers an overview of current limitations and future challenges of process mining in the context of audits whereas [10] illustrates opportunities of online auditing. [11–13] focus on fraud and outline possibilities of process mining for fraud detection and auditing thereby highlighting the potential of process mining as a new toolkit for internal audits. [5], [14] developed a mining algorithm that is able to exploit the structure of financial journal entries for the purpose of process mining in the context of financial audits. [15] Further introduce automated audit methods for testing application controls in ERP systems. [2], [3] finally conceptually describe how process mining techniques for financially relevant business processes can be combined with methods for automated control testing.

For the purpose of the research work of this paper we implemented the mining algorithm introduced by [5], [4]. Their mining technique includes the extraction of financially relevant information of journal entry values that are relevant for the purpose of auditing.

An alternative approach is used by [16]. They provide an interesting case study about the examination of mined instances of a procurement process. Their approach differs from the research presented in this paper as they actually perform a deviation analysis of the mined process instances with a manually evaluated ideal process. They base their analysis on a predefined set of process instances. The aim of the research work illustrated in this paper does not focus on providing a case study for auditing mined business processes but intends to reveal general possibilities and limitations to discover and analyze business processes from event log data without further knowledge of the underlying processes in the context of financial audits. As illustrated in [3] the ultimate aim is to develop methods that show which processes are mirrored in the available logs and how they affect the financial statements.

For analyzing mined process instances these need to be modeled in a purposeful modeling language. [17] Suggest using a BPMN representation for mined process instances in the context of financial audits. Although BPMN process models might be easier to interpret for end users we have chosen Petri nets as a modeling language for the research presented in this paper. On the one hand a broad variety of the aforementioned research work from the field of process mining relies on Petri nets as the choice of modeling language [18]. And referring to Petri nets opens up the opportunity to incorporate these already existing research results and techniques for the purpose of mining and analyzing. On the other hand Petri nets have a mathematical foundation and offer a formal graphical notation. These characteristics allow the development of sophisticated analysis methods. They therefore constitute the preferred modeling language for the research outlined in this paper. In the context of this paper we primarily refer to the publications of [19] and [20] concerning the theoretical foundation for the application of Petri nets.

3 Research Methodology

The research presented in this paper follows a design science approach [21–23]. A common critic in the academic arena refers to the perceived lack of rigor concerning design science oriented research. In order to address this aspect we have obtained extensive test and real life data for testing the designed artifacts. Actually the key aspect of this paper is to illustrate the results of evaluating already designed methods against this data. The illustrated work follows a research process as suggested by [23] consisting of the phases analysis, design, evaluation and diffusion. The requirements for an adequate representation and modeling of mined processes were investigated by considering specific, already existing literature [17] and by analyzing available test and real life data. The used mining methods were engineered [24] by assembling parts of already available methods and by developing new concepts where no adequate solutions were available yet.[1] The analysis results and requirements for further development constitute the primary outputs produced in the research process that is laid out in this paper. The engineered methods were implemented in a software prototype for evaluation purposes. We rigorously tested the software artifact with test and real life data in order to validate it against the relevant research questions addressed by the research work [25]. The content of this article discusses the results and insights that have been generated by applying the designed methods to this voluminous data.

4 Representation and Experimental Setup

[5] Introduced a simple, deterministic and unsupervised mining algorithm that is suitable for extracting data from information systems and for reconstructing executed process instances. When using process mining in the context of financial audits it is necessary to mine information that is relevant from an audit perspective and to ensure that the received information precisely reflects the executed transactions. Financial transactions in ERP systems create journal entries when they are executed. The chosen algorithm exploits the open-item-accounting structure of journal entries that can be used to link transactions to a process instance.[2] Journal entries consist of an accounting document and at least two entry items that are posted as credits and debits. When open-item-accounting is enabled each cleared item has a reference to the accounting document that cleared it. The algorithm starts with an arbitrary journal entry and reconstructs the links between journal entries that cleared each other. It matches the events in the event log to cases that represent process instances.[3]

The original mining algorithm produced directed graphs representing the mined process instances. We extended the mining algorithm with a function for mapping the

[1] The engineering of the applied methods is not part of this paper. Details are available in [5].

[2] The open-item-accounting is a fundamental concept of the double-entry bookkeeping which needs to be supported by every information system used for double-entry bookkeeping.

[3] Compared to other mining algorithms like the α-Algorithm [4] the implemented algorithm does not rely on the temporal ordering of events but on their logical structure.

mined cases to Petri nets and implemented it in a software artifact. The software prototype was written in Java using the Java NetBeans IDE [26]. It provides functionality to export Petri net models in different data formats for visual representation. The open source software Renew [27] was used for verifying that the software artifact reconstructs reachable and therefore correct Petri nets. The yEd Graph Editor [28] was used for graphical representation and automatic layout of mined process instances.

Figure 1 displays a colored Petri net (CPN) model of a reconstructed process instance. The example shows an instance of a purchasing process. Executed transactions are modeled as rectangles (Petri net transitions). The journal entry items produced by executing the transactions are modeled as circles (Petri net places). The places are colored by the account number according to the account the item was posted to. Two different types of connections are possible between transactions and journal entry items. A dotted arrow (Petri net arc) means that a transaction has posted the connected journal entry item. A dotted line (Petri net test arc) illustrates that a journal entry item was cleared by the connected transaction. Arc inscriptions play a significant role. They denote the values that are associated to the connection between transactions and journal entry items. Each transition is accompanied by a start place containing a token colored with the original document number of the journal entry. They are connected to the corresponding transaction with a simple arrow (Petri net arc). This actually leads to an enabled CPN that mimics the behavior of the originally executed process instance.

The example in Figure 1 shows that a transaction for receiving goods (MB01) was processed by user 2. It led to entries on the raw material account (310000) and on the goods received / invoices received account (191100) with the amount of 17,874.76. The invoice for these received goods were processed (MR1M) and a payment run (F110) executed that cleared the items posted by the MR1M transactions. The FB1S transaction was executed for clearing the items that were created by MB01 and MR1M.

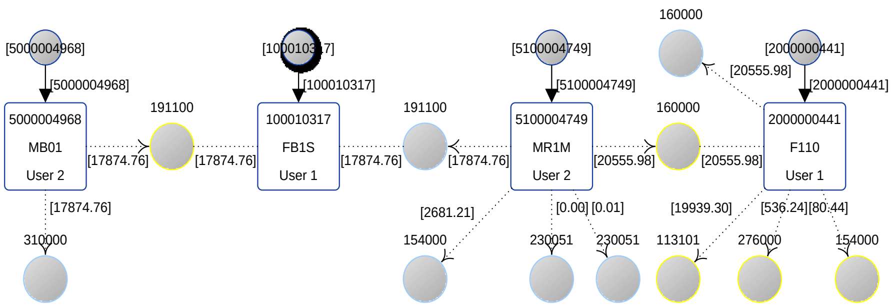

Fig. 1. Example of a Reconstructed Purchase Process Instance

For applying the mining algorithm the necessary data of the executed process instances was extracted from the available ERP systems. The setup of the experiment including conducted activities, involved software modules and input and output for each activity is illustrated in Figure 2. The relevant data was extracted by using a configurable extraction module. It retrieves the event log from the ERP system by extracting data from relevant database tables. The usage of a separate module provides the benefit that only the extraction component needs to be adjusted when data is extracted from different ERP systems. The mining algorithm operates independently from the underlying data structure of the individual ERP systems. The extraction module loads the extracted data into an event log database that can be accessed by the mining module. The mining module matches events in the log to cases, reconstructs executed instances and provides functionalities for analyzing them. It also produces output files in different formats (Extensible Graph Modeling Language (XGML) and Petri Net Markup Language (PNML)) that can be imported into subsequent software for verification (Renew) and graphical presentation (yEd Graph Editor) purposes. The reconstructed instances are modeled as Petri nets and stored as separate data objects in the mining module.

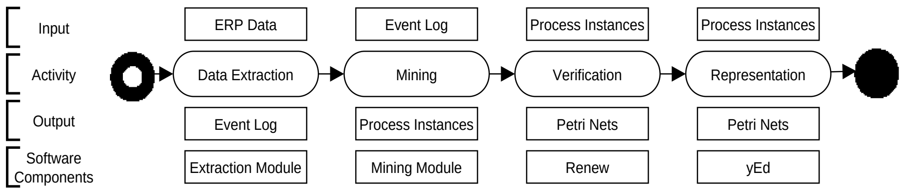

Fig.2. Experimental Setup

We used three different data sets for analysis. The first set was extracted from the SAP IDES test system. The test system is available for universities participating in the SAP University Alliance Program [29]. Postings in ERP systems are stored as data entries with information relating to the whole posting (journal entry) and the single entries that were posted to different accounts (journal entry items). The data set contained 115,060 journal entries and 419,106 journal entry items. 81,171 process instances could be reconstructed by executing the implemented mining algorithm. The data set included all transaction data that was available in the test system covering a period of 17 years.

The second data set was extracted from a SAP system of a retail company. The set included the data of all executed transactions but only for a time period of one year. The volume of 92,487 journal entries and 222,901 related journal entry items that can be traced back to 40,130 process instances over just one year illustrates the high amount of transactions that are processed in real life environments.

This observation becomes even more evident for the third data set. It originated from a SAP system of a manufacturing company in the health sector. It contains 1,764,773 journal entries and 7,395,434 journal entry items. 1,035,805 process instances could be reconstructed using the mining algorithm. Table 1 provides an overview of the different data sets.

Table 1. Overview of Data Sets

Data Set	#1 SAP IDES	#2 Retail	#3 Manufacturing
Number of journal entries	115,060	92,487	1,764,773
Number of journal entry items	419,106	222,901	7,395,434
Number of process instances	81,171	40,130	1,035,805
Covered period	17 years	1 year	1 year

5 Mining Results Analysis

The mining and reconstruction of process instances from the available data sets provide the basis for analyzing the created Petri net models. The aim of this analysis is the identification of patterns that might help in further improvement of the mining algorithm, the gaining of insights concerning which further research is needed for developing automated analysis methods that can be applied in real life scenarios, and what kind of limitations for developing such methods might exist.

The following sub-sections illustrate results from statistical analyses of the mined process instances for all three used data sets. Due to place restrictions we limit the presentation of results to those aspects that we consider relevant for the aforementioned aim.

5.1 Distribution of Net Size

Figures 3, 4 and 5 show the distribution of the number of process instances over the number of net elements with logarithmic scaling on the x- and y-axes for the different data sets.[4] Net elements include transitions, places and arcs. The number of net elements gives an impression of the size and complexity of a process instance.

The charts illustrate that the distribution of the number of net elements over the number of instances exhibit the same pattern for all data sets. Only very few instances consist of very many net elements. The vast majority of instances consist of relatively few net elements. Table 2 provides an overview of specific characteristic values of the distributions.

The shown distributions by themselves do not allow drawing conclusions concerning the design of automated business process analysis procedures. But the observation that the majority of instances actually consist of relatively few elements and that this is the case for all evaluated data sets constitutes a useful insight in combination with the analysis results highlighted in the following sub-sections.

[4] The diagrams do not show values for instances with less than three net elements. Each net consists at least of one transaction represented by a transition in the model. Each transition is accompanied by a start place that is connected to the transition and that enables the transition to fire. Therefore the minimum number of net elements per net is three.

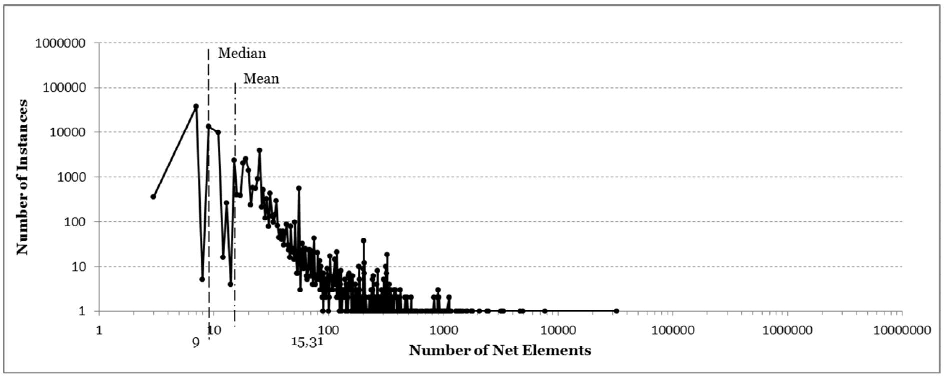

Fig. 3. Data Set #1 Distribution of Number of Net Elements over the Number of Instances

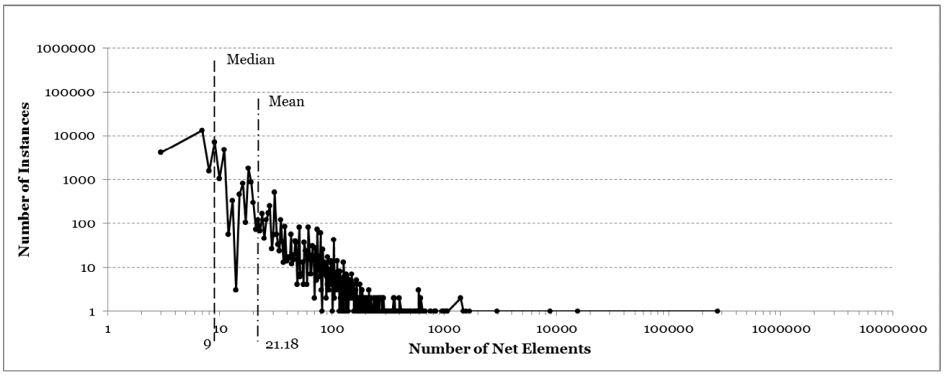

Fig. 4. Data Set #2 Distribution of Number of Net Elements over the Number of Instances

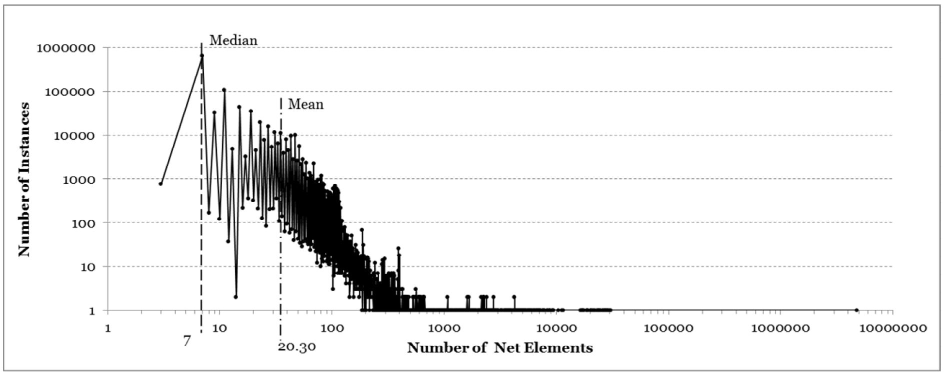

Fig. 5. Data Set #3 Distribution of Number of Net Elements over the Number of Instances

Table 2. Overview of Net Size Distribution Characteristics

Data Set	#1 SAP IDES	#2 Retail	#3 Manufacturing
Mean value of net elements per instance	15.31	21.28	20.30
Median value of net elements per instance	9	9	7
Maximum of net elements per instance	32,519	275,870	4,769,379
Standard deviation of number of net elements	127.83	1,380.41	4,688.30

5.2 Distribution of Transaction Code Combinations

Figures 6, 7 and 8 show the distribution of transaction code combinations over the number of instances. The y-axes follow a logarithmic scaling. Each number on the x-axis represents a transaction code combination. This means for example that the transaction code combination number 27 (FB1S, MB01, F110, FB05, MIRO) was executed in 1,069 process instances in data set three. These instances did not include any other transaction codes.

The distributions for all data sets show the same pattern. The majority of instances only contain very few different transaction code combinations. Taking into account the results from analyzing the distribution of net sizes from the previous section it is reasonable to assume that the majority of instances are very limited in size and reveal the same transaction code combinations.

A clustering of instances that exhibit the same size and the same transaction code combination could be a starting point for automatically analyzing a large amount of the mined instances. Each cluster could be reviewed for the value that it contributes to the financial statements and if the cluster constitutes a material process flow that needs to be further evaluated from a materiality perspective. Such an analysis would provide useful information to the auditor about how the processes in a company actually affect the financial statements.

The clustering of isomorphic graphs into clusters would also enable to search for application controls that affect the cluster under review. In combination with automated application control testing the isomorphic process instances in a cluster could be automatically audited.

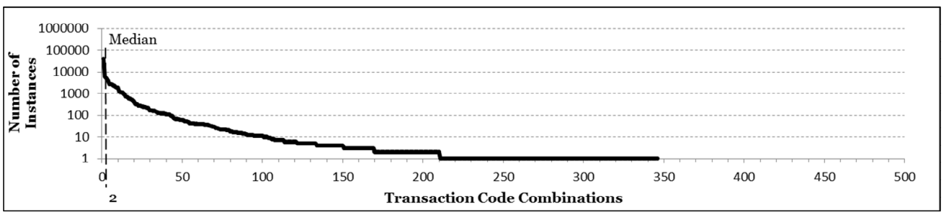

Fig. 6. Data Set #1 Distribution of the Number of Instances for Different Transaction Code Combinations

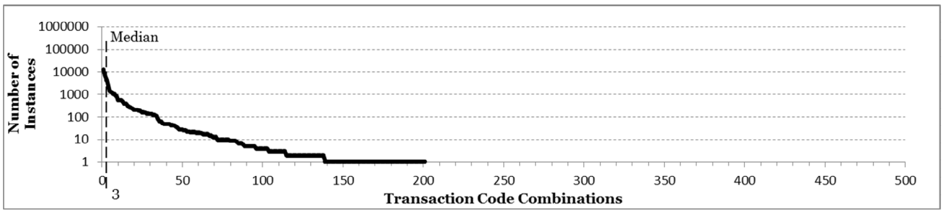

Fig. 7. Data Set #2 Distribution of the Number of Instances for Different Transaction Code Combinations

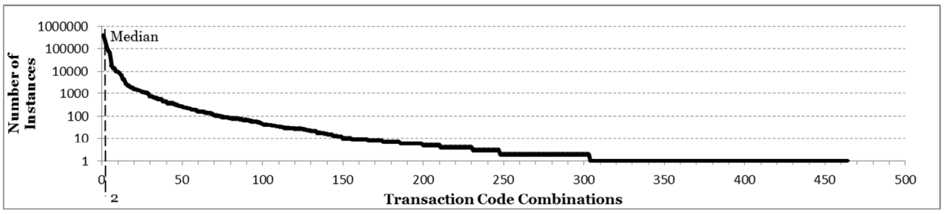

Fig. 8. Data Set #3 Distribution of the Number of Instances for Different Transaction Code Combinations

5.3 Distribution of Accounts

We suggested in the previous section that clustering could be a promising solution for analyzing similar and small instances. This leads to the question how large and complex instances should be handled. The instance in Figure 1 contains four transitions and consists of 38 net elements. An instance of this size and complexity can be evaluated by simple observation. This is not the case anymore for more complex instances. Figure 9 shows a process instance containing 3,057 transitions and consisting of 15,319 net elements. These kinds of instances cannot be evaluated without further consideration.

Figures 3 to 5 show that the number of net elements extending 100 net elements is relatively small in all the data sets. Only 691 instances in data set one consist of 100 or more net elements, 428 in data set two and 7,470 in data set three. Although the number of complex instances is relatively low they might represent a material amount of transactions that affect the financial statements and can therefore not be neglected. For analyzing and evaluating these instances it is necessary to reduce

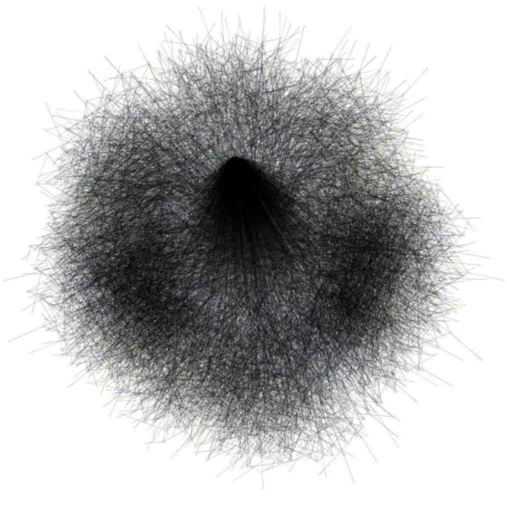

Fig. 9. Complex Process Instance

their complexity. The analysis of the accounts that are actually used within an instance provides a starting point for investigating complexity reduction possibilities. The distributions of the accounts over the number of process instances in Figures 10 to 12 show on how many accounts journal entry items were posted to in a single process instance. For example 222 instances in data set two used seven accounts to post items.[5]

The distributions for all three data sets again display the same pattern. Table 3 provides an overview of characteristic values of these distributions. The maximum number of used accounts is relatively low compared to the maximum net sizes. For interpreting the illustrated distributions and their characteristic values it is necessary to consider that only one instance in data set two uses the maximum number of 596 accounts and one in data set three the maximum of 399 accounts. All other instances do not use more than 63 accounts in data set two and no more accounts than 45 in data set three.

The observation of the distributions reveals that journal entry items are posted to relatively few accounts. This is reasonable because a specific process generally only uses a subset of the available set of accounts. The execution of a purchase process for example would likely lead to journal entries on expense accounts but not on sales accounts.

Based on the observation that the number of used accounts is relatively small it might be useful to aggregate journal entry items that were posted to the same accounts and thereby reducing the net size and complexity. Items are modeled as places in the Petri net and colored by the account number reflecting the financial account the item was posted on. The places carrying the same color (account number) could be folded leading to Petri net models that contain significantly less net elements and therefore a reduced complexity. Further research would actually be needed for verifying if a sufficient complexity level can be reached by this approach.

Table 3. Overview of Account Distribution Characteristics

Data Set	**#1 SAP IDES**	**#2 Retail**	**#3 Manufacturing**
Mean value	2.95	2.30	2.33
Median value	2	2	2
Maximum value	36	596	399
Standard deviation	1.61	3.37	0.92

[5] A logarithmic scaling is again used for the x- and y-axes.

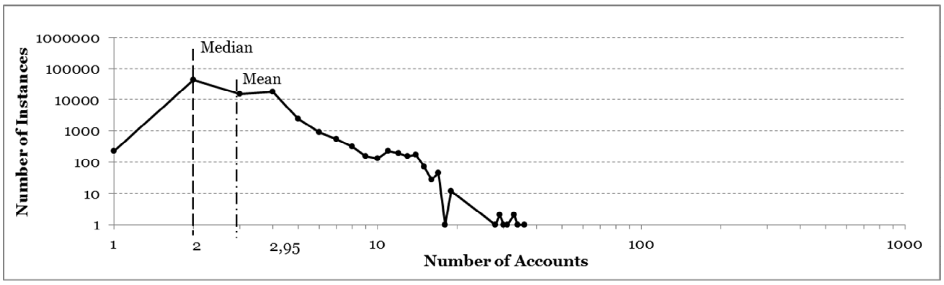

Fig. 9. Data Set #1 Distribution of the Number of Instances over Number of Accounts

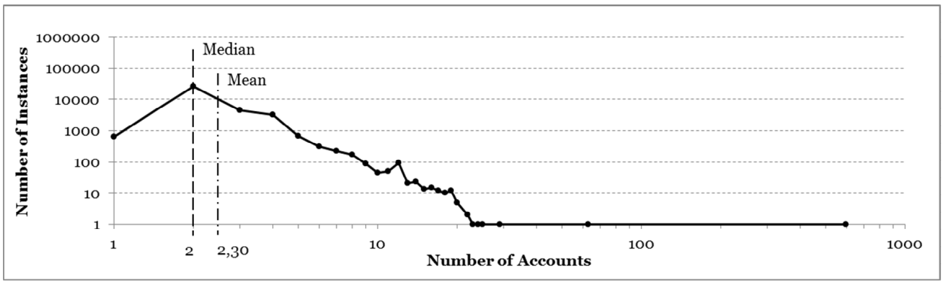

Fig. 10. Data Set #2 Distribution of the Number of Instances over Number of Accounts

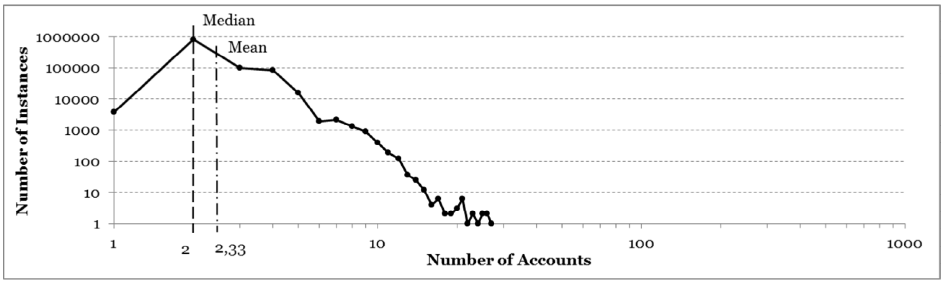

Fig. 11. Data Set #3 Distribution of the Number of Instances over Number of Accounts

6 Discussion and Limitations

The previous sections lay out insights that can be derived from analyzing results from the application of a mining algorithm. When considering these results it is necessary to keep in mind that they only provide starting points for further research and that no designed methods and their implementations do exist yet that could prove or disprove the postulated assumptions.

A further restriction is the number of used data sets. [16] Relied on an event log representing 26,185 procurement process instances from a single company in the financial sector. We therefore consider the size of the data sets as being sufficient and we also did not limit our analysis to a specific type of business process but we actually only cover two industries – retail and manufacturing. We do not know if the identified characteristics also hold true for other industries.

As a result of the analysis of used accounts we proposed the folding of places in a Petri net carrying the same account number for complexity reduction purposes. From an audit perspective it is crucial that the received information exactly mirrors the transaction that really took place and therefore to maintain the audit trail [30]. Applying a folding approach would actually lead to a graph transformation of the mined process model and it would need to be ensured that the behavior of the net remains stable [17].

A further limitation relates to the execution of the programmed software and the time that is needed to calculate the mined instances. Dedicated research on the performance of the implemented software prototype is currently outstanding but our experience from analyzing the data sets presented in this paper lets us assume that the complete reconstruction of all process instances might be unrealistic in a real life scenario and that a sampling approach based on a materiality perspective might be needed.

7 Summary and Conclusion

Research on process mining has advanced and matured significantly over the past two decades and is now increasingly applied to specific application domains. The focus of the research presented in this paper lies on the domain of financial audits. The audit of business processes is a mandatory step in the auditing process that becomes increasingly challenging with the ongoing integration of information systems and automation of transaction processing. The application of process mining for supporting the auditor comprises a promising alternative to counter the growing complexity and amount of processed transactions that need to be evaluated during an audit.

We implemented a mining algorithm in a software artifact and evaluated it against voluminous test and real life data. Based on the results derived from the analysis of reconstructed process instances we gained insights that can be used for developing automated process analysis methods in the context of financial audits.

The majority of process instances is small and consists of only few net elements. The clustering of instances that exhibit the same size and the same transaction code combination could be a starting point for automatically analyzing a large amount of the mined instances. Each cluster could be reviewed for the value that it contributes to the financial statements. Such an analysis would provide useful information to the auditor about how the processes in a company actually affect the financial statements.

Process instances containing more than a few processed transactions cannot be evaluated manually by simple observation. Although the overall number of complex process instances is very limited they cannot be neglected from an audit perspective. A single instance may already contain extremely high volumes of transactions that might be material. A starting point for reducing complexity could be the consideration of used accounts. Even for large and complex process instances the number of used accounts in a single process instance is relatively small. The folding of equally colored places in process instances could lead to significant complexity reduction especially for large process instances.

The research presented in this paper can be seen as a step towards the development of automated business process analysis methods. The accounting scandals of major companies over the last years illustrates that the audit industry is currently lacking adequate solutions for safeguarding the correctness of published financial statements. The usage of automated analysis and audit methods constitutes a necessary requirement to overcome the existing imbalance between automated processing on the companies' side and manual audit procedures on the auditors' side. The introduction of automated audit procedures has the potential to leverage this imbalance.

Limitations concerning the applicability of the implemented algorithm exist especially in regard to processing time and the amount of process instances that can be mined and analyzed. At this point of time it also remains unclear if the identified starting points can actually be transferred successfully into the design of automated business process analysis methods. But first results from consecutive research that bases on the results presented in this paper that will be published in forthcoming articles provide a positive indication.[6]

References

1. International Federation of Accountants: ISA 315 (Revised), Identifying and Assessing the Risks of Material Misstatement through Understanding the Entity and Its Environment, (2012)
2. Werner, M., Gehrke, N.: Potentiale und Grenzen automatisierter Prozessprüfungen durch Prozessrekonstruktionen. Forschung für die Wirtschaft. Shaker (2011)
3. Werner, M., Gehrke, N., Nüttgens, M.: Business Process Mining and Reconstruction for Financial Audits. Hawaii International Conference on System Sciences, pp. 5350–5359. Maui (2012)
4. Van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin/ Heidelberg (2011)
5. Gehrke, N., Müller-Wickop, N.: Basic Principles of Financial Process Mining A Journey through Financial Data in Accounting Information Systems. Proceedings of the 16th Americas Conference on Information Systems. Lima (2010)
6. Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. Proc. Sixth Int'l Conf. Extending Database Technology, pp. 469–483 (1998)
7. Musaji, Y.F.: Integrated Auditing of ERP Systems. John Wiley & Sons (2002)
8. Alles, M., Jans, M., Vasarhelyi, M.: Process Mining: A New Research Methodology for AIS. CAAA Annual Conference 2011 (2011)
9. Jans, M.J.: Process Mining in Auditing: From Current Limitations to Future Challenges. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.): Business Process Management Workshops, pp. 394–397. Springer, Berlin/Heidelberg (2012)

[6] The presented results were developed in the research project Virtual Accounting Worlds. The project is sponsored by the German Federal Ministry of Education and Research (grant number 01IS10041). The authors are responsible for the content of this publication.

10. Aalst, W.M.P. van, Hee, K.M. van, Werf, J.M. van, Verdonk, M.: Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. Computer. 43, pp. 90–93 (2010)
11. Jans, M., Lybaert, N., Vanhoof, K., Van Der Werf, J.M.: Business process mining for internal fraud risk reduction: Results of a case study. (2008)
12. Jans, M., Alles, M., Vasarhelyi, M.: Process mining of event logs in auditing: opportunities and challenges. Working paper. Hasselt University, Belgium (2010)
13. Jans, M., Van der Werf, J.M., Lybaert, N., Vanhoof, K.: A business process mining application for internal transaction fraud mitigation. Expert Systems with Applications. 38, pp. 13351–13359 (2011)
14. Gehrke, N., Müller-Wickop, N.: Rekonstruktion von Geschäftsprozessen im Finanzwesen mit Financial Process Mining. Lecture Notes in Informatics, Proceedings der Jahrestagung Informatik. Leipzig (2010)
15. Gehrke, N.: The ERP Auditlab-A Prototypical Framework for Evaluating Enterprise Resource Planning System Assurance. 43rd Hawaii International Conference on System Sciences (HICSS) (2010)
16. Jans, M., Alles, M., Vasarhelyi, M.: Process Mining of Event Logs in Internal Auditing: A Case Study. 2nd International Symposium on Accounting Information Systems (2011)
17. Müller-Wickop, N., Schultz, M., Gehrke, N., Nüttgens, M.: Towards Automated Financial Process Auditing: Aggregation and Visualization of Process Models. Proceedings of the Enterprise Modelling and Information Systems Architectures. Germany (2011)
18. Tiwari, A., Turner, C., Majeed, B.: A review of business process mining: state-of-the-art and future trends. Business Process Management Journal. 14, pp. 5–22 (2008)
19. Van der Aalst, W.M.P., Stahl, C.: Modeling business processes : a petri net-oriented approach. MIT Press, Cambridge, Mass. (2011)
20. Valk, R.: Lecture Notes: Formale Grundlagen der Informatik II (FGI 2) Modellierung & Analyse paralleler und verteilter Systeme. University of Hamburg (2008)
21. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly, pp. 75–105 (2004)
22. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision support systems. 15, pp. 251–266 (1995)
23. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on design-oriented information systems research. European Journal of Information Systems. 20, pp.7–10 (2010)
24. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology. 38, pp.275–280 (1996)
25. Riege, C., Saat, J., Bucher, T.: Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik. Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik, pp.69–86 (2009)
26. Oracle: Welcome to NetBeans, http://netbeans.org/
27. University of Hamburg: Renew - The Reference Net Workshop, http://www.renew.de/
28. yWorks GmbH: yEd - Graph Editor, http://www.yworks.com/de/products_yed_about.html
29. SAP: SAP-UCC, http://www.sap-ucc.com/
30. Romney, M.B., Steinbart, P.J.: Accounting Information Systems. Prentice Hall (2008)